Bedienungsanleitung

tv-computersystem 6800



Bedienungsanleitung

tv-computersystem

6800



Fernsprecher (07657) * 444
Fernschreiber 07 722 323 fmkg d

Oktober 1976   2. verbesserte Auflage

Bedienungsanleitung

TV-Mikroprozessorsystem 6800

Inhalt: Seite

1. Betriebsanleitung TV-Computersystem 6800 Seite

1. Betriebsanleitung TV-Computersystem 6800

1.1. Einleitung

Bei komplexen Geräten steigt die Wahrscheinlichkeit der fehlerhaften Bedienung. In besonderem Maße trifft dies auf die Programmierung von Computern zu. Glücklicherweise zerstört eine fehlerhafte Bedienung den Computer nicht. Die Eigenschaft des Computers, nur fehlerfreie Programme richtig zu verarbeiten und die Neigung Vieler, bei einem nichtfunktionierenden Programm die Schuld beim Computer und nicht bei sich selbst zu suchen, zwingt zu einer Arbeitsweise, die es ermöglicht, das Ziel - ein fehlerfreies Programm - mit gutem Wirkungsgrad zu erreichen.

In entsprechendem Maße betrifft dies auch die Handhabung des Gerätes. Deshalb empfehlen wir, die Bedienungsanweisung gründlich durchzuarbeiten und sich mit Hilfe des Gerätes die Bedienung einzuprägen. So lassen sich Mißverständnisse und Rückfragen vermeiden.

Um das Studium der vorwiegend englischen Literatur nicht zu erschweren, war es nicht sinnvoll, alle englischen Ausdrücke zu übersetzen.

Besonders die mnemotechnischen Befehlsbezeichnungen entsprechen der ursprünglichen Fassung. Die Lesbarkeit schon geschriebener Programme läßt hierzu keine Wahl. Im Kapitel 15 sind die englischen Ausdrücke übersetzt und erklärt.

1.2. Geräte-Kurzbeschreibung

Der Schulungsrechner "TV-Computersystem 6800" besteht aus einem Mikroprozessor (6800 von Motorola bzw. AMI) mit Speicher und einer Elektronik, die den Speicherinhalt auf einem FS-Gerät abbildet. Mittels Lichtgriffel kann durch die zeitliche Koinzidenz ein Programm in Maschinensprache unmittelbar in den Speicher übertragen, gestartet und das Ergebnis auf dem Bildschirm kontrolliert werden. Als Ein- und Ausgabemedium dient in beiden Fällen der Bildschirm eines handelsüblichen Fernsehgerätes, das über den Antenneneingang angesteuert wird. Dadurch entfallen aufwendige Peripheriegeräte.

Die Programme bzw. Daten werden bit- oder tetradenweise durch den optischen Kontakt des Lichtgriffels mit den, auf dem Bildschirm dargestellten Steuerfeldern, in das verfügbare 1 k Byte RAM des Mikroprozessors eingegeben. Dabei wird durch eine entsprechende Anzeige jede Eingabe quittiert. Die Ausgabe erfolgt ebenfalls in Bit- oder Tetradendarstellung auf dem Bildschirm.

Für die Ein- und Ausgabe zeigt der Bildschirm einen Speicherausschnitt von 32 Bytes, die durch 64 Tetraden oder 256 Bit dargestellt sind. Die 32 möglichen Speicherausschnitte lassen sich durch Steuerfelder am rechten Bildrand neben dem Speicherfeld über einen Vorwärts-/Rückwärts-Zähler oder über Tetraden anwählen.

Ein Steuerfeld schaltet zwischen der tetraden- und bitweisen Darstellung des Speicherfeldes um. Ein weiteres ermöglicht das Überschreiben des dargestellten Speicherausschnitts mit einer beliebig anwählbaren Tetrade zwischen 0 und F. Weitere vier Felder stehen für die Mikroprozessorsteuerung zur Verfügung:

Grundzustand - Start

Halt - Fortstart

Einzelbefehl mit Registeranzeige

Interrupt

Das Programm und damit die Startadresse kann beliebig im Speicher plaziert werden. Beim Einzelbefehl sind im ebenfalls beliebig plazierbaren "Stack" die Inhalte aller Register nach der Befehlsausführung abgelegt. Über den Interrupt läßt sich das laufende Programm unterbrechen und eine Interrupt-Routine starten.

1.3. Anschlußbedingungen

Das Gerät läßt sich an jeden handelsüblichen Schwarzweiß-Fernsehempfänger anschließen. Farbfernseher sind wegen ihrer internen Verzögerung nicht geeignet. Sinnvollerweise sollte die Bildschirmdiagonale nicht kleiner als 30 cm sein. Es besteht sonst die Gefahr des unbeabsichtigten Setzens von Nachbarpunkten.

Zweckmäßig steht das Gerät so auf dem Fernsehempfänger, daß der Lichtgriffel rechts am Bildschirm herunterhängt. Ist dies nicht möglich, kann das Gerät auch links vom FS-Gerät plaziert werden.

Das Antennenkabel wird mit der Koax-Antennenbuchse des FS-Empfängers verbunden. Sollte das Gerät noch mit Bananensteckeranschlüssen versehen sein, ist ein entsprechendes Zwischenstück erforderlich. Dieses kann bezogen werden. Nach Verbinden mit dem Netz und Einschalten beider Geräte, ist der FS-Empfänger auf Band 3, Kanal 7 einzustellen.

Bei nicht genauen Kanalangaben am Fernsehgerät finden Sie den "TV-Computer-Sender" ungefähr in der Bandmitte.

Das Gerät ist jetzt betriebsbereit.

1.4. Funktionserklärung der Bedienungselemente

1.4.1. Bedienungselemente Gerät

Auf der Geräterückseite befinden sich, je nach Ausführung, 2 Anschlußbuchsen: Eine ist für den Anschluß eines Kassettenrekorders vorgesehen, die andere führt den Bus und die Mikroprozessorsteuersignale nach außen. Näheres hierzu finden Sie im Kapitel 17: "Anhang."

1.4.2. Bedienungselemente Bildschirm

Die Seite 61 der Bedienungsanleitung zeigt die Funktion der Steuer- und Anzeigefelder.
Auf der rechten Bildseite befinden sich die Bedienungsfelder.
Die linke Bildseite und Bildmitte zeigt einen Ausschnitt des Speichers. Dieser Ausschnitt kann bit- oder tetradenweise dargestellt werden. Die Adressenreihenfolge wird auf dem Bildschirm zeilenweise dargestellt. Der jeweils gezeigte Ausschnitt wird im folgenden als Seite bezeichnet.

Bei der bitweisen Darstellung ist jedem Bit ein Steuerfeld zugeordnet. Bei optischem Kontakt des Lichtgriffels mit dem Steuerpunkt wird der entsprechende Speicherinhalt alle halbe Sekunde invertiert. Er läßt sich damit leicht setzen und löschen.

Bei der Tetradendarstellung bilden jeweils vier Bit eine Tetrade. Diese Tetraden können ebenfalls mittels Lichtgriffel verändert werden. Die zu setzende Tetrade ist jeweils am rechten Bildrand auszuwählen.

Die Bedienungsfelder sind in zwei vertikalen Reihen angeordnet. Die linke Reihe zeigt neben den 16 Steuerfeldern die Symbole "0" bis "F". Über die daneben stehenden Steuerfelder läßt sich die entsprechende Tetrade durch Antippen mit dem Lichtgriffel anwählen. Im Anzeigefeld rechts des Symboles "4" erscheint die angewählte Tetrade. Die Symbole "0" bis "F" dienen auch zur Kennzeichnung der Zeilen.

Die Steuerfelder in der rechten Spalte dienen als Schalter. Mit Ausnahme der Seitenzählersteuerung rechts der Felder "4" bis "7", zeigen sie den eingeschalteten Zustand durch ein helles Feld an.
Über das Steuerfeld rechts der "0" läßt sich die dargestellte Speicherseite mit der angewählten Tetrade füllen, vorausgesetzt, die Speicherseite wird durch Tetraden dargestellt. Danach ist der Schalter wieder auszuschalten.

Das Steuerfeld rechts der "2" schaltet zwischen der Bit- und Tetradendarstellung um.

Die Felder neben der "4" und "5" dienen zum Vorwärts- und Rückwärtszählen des Seitenzählers. Der Zählerstand steht in den Feldern neben der "6" und "7". Über die dortigen Steuerfelder läßt sich jede Stelle des Zählers auch mit der angewählten Tetrade setzen. Der höchste Wert ist 1F und entspricht der zweiunddreißigsten Seite.

Die vier Steuerfelder neben der "9", "B", "D" und "F" dienen der Mikroprozessorsteuerung. Sie bewirken in gleicher Reihenfolge:
Interrupt, Einzelbefehl mit Registeranzeige, Halt-Fortstart, Grundzustand-Start.

Diese vier Felder stehen nach dem Einschalten des Gerätes auf "aus".

Nach dem Aufheben des Grundzustandes durch das Einschalten des Start-Feldes läuft das im Speicher stehende Programm. Der Programmlauf ist durch die Halt-Steuerung unterbrechbar. Während der Halt ansteht, kann an der unterbrochenen Stelle mit der Einzelbefehlsteuerung fortgefahren werden. Dabei können alle Registerinhalte in einem frei wählbaren Speicherbereich angezeigt werden. Über den "Interrupt-Schalter" läßt sich das gerade laufende Programm unterbrechen. Es läuft dann ein Interrupt-Programm ab.

1.4.3. Übersicht über die Bildschirmbedienungselemente

Steuerfelder

Steuerfelder sind kleine helle Quadrate, die zur Lichtgriffeleingabe dienen, sie befinden sich links oben neben den Anzeigefeldern.

Anzeigefelder

Die rechts neben den Steuerfeldern plazierten Anzeigefelder zeigen den Schaltzustand des Feldes an. Dieser wird binär durch das Vorhandensein eines Rechteckes oder dessen Fehlen, oder tetradenweise durch ein Symbol zwischen "0" und "F" dargestellt.

Speicherseite

Der auf dem Bildschirm in Bit oder Tetradenform dargestellte Speicherausschnitt. Jeder Ausschnitt hat 16 x 16 Bit oder 4 x 16 Tetraden = 32 Bytes. Jedes Speicherelement, Bit oder Tetrade, besteht aus einem Steuer- und einem Anzeigefeld. Die Nummer der angezeigten Seite steht auf der rechten Bildseite rechts der "6" und "7".

Speicher (RAM)

Der Speicher besteht aus 1k Byte = 1024 Worte zu je 8 Bit. Dies entspricht 32 Speicherseiten. Die Speicherseiten sind von "0" bis "1F" bezeichnet. Der Adressenbereich geht von 0 bis 3FF. Siehe auch Speicherbelegungsplan im Abschnitt 5.

Seitenzähler

Der Seitenzähler wählt die Seite aus. Über die zwei Steuerfelder rechts der "4" und "5" können die Seiten vorwärts und rückwärts durchgezählt werden. Die zwei Zählerstellen sind auch über die Steuerfelder rechts der "6" und "7" mit der angewählten Tetrade zu setzen. Der maximale Zählerstand ist 1F. Der Zählerstand steht rechts der Tetraden "6" und "7".

Tetradenauswahl

Die Tetraden von "0" bis "F" auf der rechten Bildseite haben eine Doppelfunktion. Sie bezeichnen die Zeilen und es gibt über die dazugehörenden Steuerfelder die Möglichkeit, eine Tetrade anzuwählen. Die angewählte Tetrade erscheint im Felde rechts des Symbols "4". Es läßt sich diese Tetrade an eine beliebige Stelle der Speicherseite setzen.

Seitendarstellung, Bit - Tetrade

Mit dem Steuerfeld rechts der Tetrade "2" läßt sich der Anzeigezustand der dargestellten Seite zwischen Bit und Tetrade umschalten.

Seite mit Tetrade setzen

Wenn die Seite in Tetradenform dargestellt wird, läßt sich über das Steuerfeld rechts neben dem Symbol "0" die ganze Seite mit der angewählten Tetrade vollschreiben, bei der angewählten Tetrade "0" entsprechend löschen.

Läßt man im eingeschalteten Zustand dieses Schalters den Speicherzähler einmal voll durchlaufen, so ist der gesamte Speicherinhalt gelöscht, bzw. vorbesetzt.

Grundzustand - Start

Wird dieser Schalter rechts neben "F" eingeschaltet, startet der Mikroprozessor das im Speicher stehende Programm. Die Startadresse entnimmt er der letzten Zeile der letzten Speicherseite (1F) Adresse 3FE, 3FF. Ist die Adresse Null, beginnt das Programm auf der ersten Seite in der ersten Zeile. Steht kein Programm im Speicher, bzw. stimmt die Startadresse nicht, läßt sich das Verhalten des Mikroprozessors nicht vorhersagen. Im ungünstigsten Falle überschreibt er den gesamten Speicher mit irgendeiner Information.

Halt - Fortstart

Durch dieses Steuerfeld rechts von "D" läßt sich ein laufendes Programm anhalten und fortstarten. Dies ist besonders für Testzwecke interessant.

Einzelbefehl mit Registeranzeige

Schalter rechts neben "A". Nur bei eingeschaltetem Halt kann mit Einzelbefehl fortgefahren werden. Im Stack werden dabei jeweils alle Registerinhalte abgelegt. Siehe Abschnitt 6, Beispiel. Dieses Steuerfeld dient ebenfalls Testzwecken, siehe auch Abschnitt 3, Testpraktiken. Zur richtigen Ausführung des Einzelbefehls benötigt der Mikrocomputer in der zweitletzten Zeile der letzten Seite (1F) die Adresse 03 F6. In der fünften Zeile von unten, auf der gleichen Seite, muß 02 3B stehen.

Dies hat seinen Grund in der Hardware-Ausführung des Einzelbefehls. Es wird hierzu eine Interruptmöglichkeit des Rechners ausgenützt. Das Interruptprogramm darf nur aus den Befehlen NOP und RTI bestehen. Da nach dem Rückschalten des Einzelbefehls die Information vom Stack in die Register zurück übertragen wird, kann diese Information vor dem Rückschalten verändert werden. Es lassen sich so die Register vorbesetzen.

Interrupt

Mit diesem Schalter - rechts von "9" - läßt sich ein laufendes Programm unterbrechen. Der Mikroprozessor startet ein zweites vorzugebendes Programm und fährt nach dessen Rücksprung im ersten Programm fort. Ein Anwendungsfall wäre ein dialogfähiges Programm, wobei der Regiewechsel über dieses Steuerfeld verläuft.

Ein weiterer Fall ist der programmierbare Halt. Will man an eine Adresse im Programm einen Halt einsetzen, so eignet sich hierzu besonders der WAI (warten auf Interrupt) 3E. Läuft das Programm auf diesen Befehl, werden alle Registerinhalte im Stack abgelegt. Besteht das Interruptprogramm nur aus dem Rücksprung, RTI, so läuft nach dem Betätigen des Schalters das Programm weiter.

Wie beim Einzelbefehl, ist in diesem Fall die Interrupt-Programmadresse vorzugeben. Sie steht in der vierten Zeile von unten auf der Seite (1F) Adresse 3F8, 3F9. Das Programm muß mit einem RTI 3B abschließen.

1.5. Software Betriebsbedingungen

In den letzten vier Zeilen (je 16 Bit oder vier Tetraden) der letzten Speicherseite (1F) sind vor dem Programmstart folgende Daten einzutragen:

Adressen	Speicher	Zeile
3F6, 3F7	02 3B	B (NOP und RTI)
3F8, 3F9		C maskierbarer Interrupt
3FA, 3FB		D Software-Interrupt
3FC, 3FD	03 F6	E Interrupt-Einzelbefehl-
3FE, 3FF		F Startadresse

Sind die letzten zwei Byte gelöscht, beginnt der Programmstart immer auf der Speicherzelle 0, Seite 0, erste Zeile. Zur Durchführung des Einzelbefehls ist eine Adresse in der Speicherzelle 3FC, 3FD einzutragen. Diese muß auf einen NOP und RTI, Rückkehr von Interrupt, verweisen. Der nichtmaskierbare Interrupt wird zur Ausführung des Einzelbefehls herangezogen.

2. Ratschläge für die Arbeitsweise

- Da sich der "direkte" Adressenmodus auf das erste Viertel k-Byte bezieht, ist es sinnvoll, häufig benötigte Programm- und Datenteile in diesen Bereich zu legen. Das Programm wird dadurch kürzer.

- Um für evtl. Änderungen ausreichend Platz zu finden, sollten nicht alle Seiten vollgeschrieben werden. Die letzten Bytes sind mit dem Leerbefehl "NOP" 02 aufzufüllen.

- Wichtige Programmparameter, evtl. auch der Stack, sind in Seite 0 zu legen. Sie erleichtern die Kontrolle und das Austesten.

- Voraussetzung zum effektiven Programmieren ist die gute Kenntnis der Befehle, ihrer Wirkung und der grundsätzlichen Programmierungstechniken. Sie haben die Funktion einer Sprache, in der das Lösungsverfahren zu formulieren ist. Je besser diese Sprache beherrscht wird, um so eleganter und effektiver ist die Problemlösung an die Möglichkeiten des Rechners anzupassen.

- Folgendes Vorgehen erleichtert das Arbeiten:

  1. Aufgabe möglichst genau beschreiben,
  2. Lösungsmöglichkeiten suchen, evtl. durch Zerlegen in Teilaufgaben.

  Von den meist mehreren denkbaren Lösungen sind solche mit den folgenden Eigenschaften vorzuziehen:

  - Übersichtlichkeit
  - leichte Testbarkeit - siehe Abschnitt 3
  - kleiner Aufwand

  Die gefundene Lösung sollte vor dem Codieren durch Texte und Flußdiagramme dokumentiert sein.

Zum Codieren eignen sich besonders die beigelegten Formulare. Jedes Blatt entspricht einer Seite auf dem Bildschirm. Es ist sinnvoll, zuerst im Externcode die Befehle einzutragen und den dazu später erforderlichen Platzbedarf ein, zwei oder drei Byte zu berücksichtigen. Nach dem Codieren sollte das Programm nochmals manuell Befehl für Befehl durchgespielt werden. Erfahrungsgemäß fällt dabei ein großer Teil der Fehler auf. Das zusätzliche Einprägen dabei erleichtert das folgende Austesten. Der naheliegende Einwand, der Rechner könne das "Durchspielen" viel schneller und besser, trifft zwar zu. Wer jedoch die Vielfalt der Fehlermöglichkeiten kennt und die Schwierigkeit, manche Fehlerarten einzukreisen, wird vorher versuchen, die Zahl der Fehler auf ein Minimum zu reduzieren.

Ist das Programm codiert, kann es in den Speicher übertragen werden. Zuvor sollte der ganze Speicher gelöscht sein.

Das eingeschriebene Programm ist unbedingt nochmals mit der Codierliste zu vergleichen.

Über die dann folgende Testpraxis finden sich im Abschnitt 3 einige Hinweise.

3. Testpraktiken

Es folgen einige Hinweise für den Ungeübten.
Schon bei der Programmkonstruktion wird über die leichte Austestbarkeit mitentschieden. Ein übersichtlich gegliedertes, wenig verschachteltes Programm ist leichter auszutesten.

Nach dem ersten Start eines Programmes können drei Situationen eintreten:

1. Es bringt die erwarteten Ergebnisse (wenig wahrscheinlich).
2. Es bringt falsche Ergebnisse.
3. Es bringt keine Ergebnisse.

Im ersten Fall folgen nun weitere Starts mit variierten Eingangsdaten, um das Programm in möglichst extremen Situationen auszutesten.

Im zweiten Fall werden ebenfalls mit varierten Eingangsdaten neue Starts durchgeführt. Aus der Art der fehlerhaften Ergebnisse läßt sich vielleicht auf die Fehlerursache schließen.

Im letzten Falle schließlich ist festzustellen, was tatsächlich passiert ist. Befindet das Programm sich in einer Schleife, ist dies durch den Einzelbefehlsablauf leicht zu erkennen, da aus den Registerinhalten alle wesentlichen Informationen zu entnehmen sind. Über den Einzelbefehl läßt sich ein Fehler generell einkreisen. Nur ist es zu langwierig, das ganze Programm per Einzelbefehl durchlaufen zu lassen.

Um von einer bestimmten Stelle aus mit dem Einzelbefehl fortzufahren, kann dort der Befehl WAI 3E eingesetzt werden. Dieser Befehl wirkt dann als Programm-Stop. Im Stack stehen dann die Registerinhalte. Nach Einlegen des Haltes kann mit Einzelbefehl fortgefahren werden.

Bei der Programmkonstruktion ist an wichtigen Knotenpunkten ein Leerbefehl NOP 02 vorzusehen. Dieser ist dann bei Bedarf durch den WAI zu überschreiben.

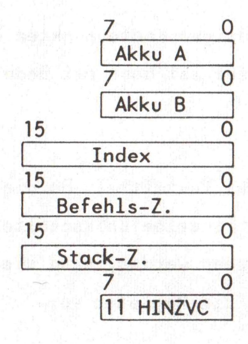
Der Einzelbefehlsablauf ist ein wichtiges Testmittel. Da die Registerinhalte nach dem Rücksetzen des Einzelbefehlsschalters vom Stack in die Register zurückgeschrieben werden, sind die Inhalte vorher auch veränderbar und damit die Register vorzubesetzen.

Eine weitere Möglichkeit, den Programmablauf zu kontrollieren, ist, an wichtigen Stellen Programmwerte, Adressen oder ähnliches zwischenzuspeichern. Aus dem Inhalt der Speicherplätze ergeben sich dann wertvolle Hinweise. Zweckmäßig liegen diese Speicherplätze in der Seite 0.

Sind zum Beheben eines Fehlers an einer Stelle einige Befehle einzufügen, so behilft man sich mit der sogenannten Rucksackmethode. An der besagten Stelle wird ein Sprung auf einen noch freien Speicherbereich eingefügt. Dort folgt der durch den Sprungbefehl überschriebene Befehl, die einzufügenden Befehle und der Rücksprung.

Generell gilt, wie für jede Problemlösung, Sammeln von Fakten, Ordnen und Auswerten.

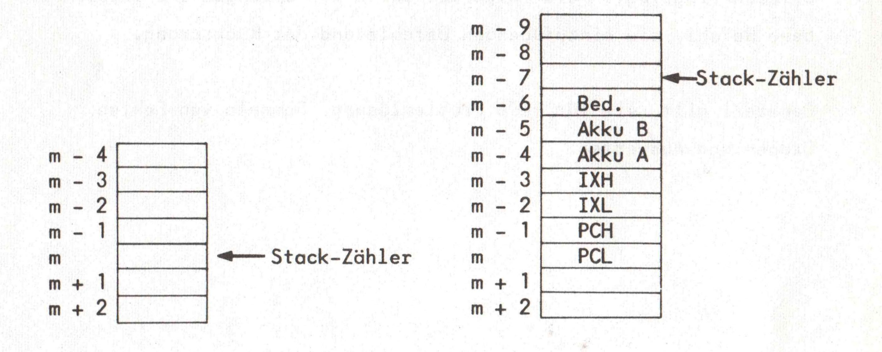
4. Register im 6800



C = Carry - Übertrag
V = Overflow - Überlauf
Z = Zero - Null
N = Negative - Negativ
I = Interrupt - Unterbrechung
H = Half Carry from Bit 3 - Halb-Übertrag von Bit 3

Abspeichern der Register im Stack

Bed.	= Bedingungscode-Register	
Akku A	= Akkumulator A	
Akku B	= Akkumulator B	
IXH	= Index-Register	höherwertiges Byte
IXL	= Index-Register	niederwertiges Byte
PCH	= Befehlszähler	höherwertiges Byte
PCL	= Befehlszähler	niederwertiges Byte



vorher nachher

5. Speicherbelegungsplan 1 k Byte

Jede TV-Seite hat 32 Bytes.

Nr.	TV-Seite	Anf.-Adr.		End-Adr.	Bemerkung
0	0	0	-	1F	
1	1	20	-	3F	
2	2	40	-	5F	
3	3	60	-	7F	0,25 k Byte
4	4	80	-	9F	"direkt"
5	5	A0	-	BF	
6	6	C0	-	DF	
7	7	E0	-	FF	
8	8	100	-	11F	
9	9	120	-	13F	
10	A	140	-	15F	
11	B	160	-	17F	
12	C	180	-	19F	0,5 k Byte
13	D	1A0	-	1BF	
14	E	1C0	-	1DF	
15	F	1E0	-	1FF	
16	10	200	-	21F	
17	11	220	-	23F	
18	12	240	-	25F	
19	13	260	-	27F	0,75 k Byte
20	14	280	-	29F	
21	15	2A0	-	2BF	
22	16	2C0	-	2DF	
23	17	2E0	-	2FF	
24	18	300	-	31F	
25	19	320	-	33F	
26	1A	340	-	35F	
27	1B	360	-	37F	
28	1C	380	-	39F	1 k Byte
29	1D	3A0	-	3BF	
30	1E	3C0	-	3DF	
31	1F	3E0	-	3FF	

6. Einfache Beispiele

Inhalt: Seite

6.1. Zählschleife

Zur schnellen Funktionskontrolle des Gerätes eignet sich folgende Zählschleife:

- o Es wird der gesamte Speicher gelöscht.
- o Es wird folgendes Programm eingeschrieben:

Seite: 0

Zählschleife

Byte	Marke	OPEX	A	Symb. Adr.	CODE		Kommentar
0		INX			0	8	Erhöhe Index + 1
1		STX	D	10	D	F	Speichere Index nach -10-
2					1	0	
3		JMP	E	00	7	E	Springe auf -00-
4					0	0	
5					0	0	
6							
7							

In der neunten Zeile - neben der Tetrade "8" zählt das Programm nach dem Start dauernd hoch.

6.2. Einzelbefehl und Interrupt

Dieses Beispiel demonstriert den Interrupt und Einzelbefehl:

Seite: 0

Einzelbefehl und Interrupt

Byte	Marke	OPEX	A	Symb. Adr.	CODE		Kommentar
0		LDS	I	1F	8	E	Lade Stack mit 1F
1					0	0	
2					1	F	
3		CLI			0	E	Lösche Interruptmaske
4		JMP	E	20	7	E	Springe auf Seite 1
5					0	0	
6					2	0	
7							

Seite 0 (Fortsetzung)

Ablage der Registerinhalte bei Einzelbefehl und Interrupt

Byte	Marke	OPEX	A	Symb. Adr.	CODE	Kommentar
⋮	⋮	⋮	⋮	⋮	⋮	⋮
18						
19						Bedingungsregister (CCR)
1A						Akkumulator B (ACCB)
1B						Akkumulator A (ACCA)
1C						Indexregister High
1D						Indexregister Low
1E						Programmzähler PC High
1F						Programmzähler PC Low

Seite: 1

Einzelbefehl und Interrupt

Byte	Marke	OPEX	A	Symb. Adr.	CODE	Kommentar
20	M1	INX			0 8	Erhöhe Index + 1
1		STX	D	0A	D F	Speichere Index nach 0A
2					0 A	
3		BRA	R	M1	2 0	Springe um -5
4					F B	
5		INX	D		0 8	Erhöhe Index + 1
6		STX		OC	D F	Speichere Index nach OC
7					0 C	
8		RTI			3 B	Rückkehr aus Interrupt
9						

Seite: 1F

Einzelbefehl und Interrupt

Byte	Marke	OPEX	A	Symb. Adr.	CODE	Kommentar
⋮	⋮	⋮	⋮	⋮	⋮	⋮
15						
16		NOP			0 2	Keine Operation
17		RTI			3 B	Rücksprg. v. Interr. E.-Bef.
18					0 0	Startadresse mask. Interrupt
19					2 5	
1A					0 0	Startadresse Softwareinterr.
1B					0 0	
1C					0 3	Einzelbefehlsinterrupt
1D					F 6	
1E					0 0	Startadresse Hauptprogramm
1F					0 0	

7. Befehlsformular

Franz MORAT KG
Elektro-Feinmechanik und Maschinenbau
7821 Eisenbach/Hochschwarzwald 1

Fernruf Eisenbach (0 76 57) * 444, 292
Fernschreiber 07 722 323 fmkg d

TV-Computersystem 6800

Seite: ....

Aufgabe: ....

Nr.	Byte	Marke	OPEX	A	Symb. Adr.	CODE	Kommentar
0	0						
1	1						
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						
8	8						
9	9						
10	A						
11	B						
12	C						
13	D						
14	E						
15	F						
16	10						
17	11						
18	12						
19	13						
20	14						
21	15						
22	16						
23	17						
24	18						
25	19						
26	1A						
27	1B						
28	1C						
29	1D						
30	1E						
31	1F						

Seite = TV-Bild
Byte = Nr. auf Seite
OPEX = externer Code

A = Adressenmodus
CODE = interner Code

A: Immediate = I, Implied = M,
Extended = E, Relativ = R,
Index = X, Accu. = A oder B

8. Rechnerstruktur

Der 6800 ist ein 8 Bit Mikroprozessor. Er ermöglicht den maskierten und unmaskierten Interrupt und die Halt-, Fortstart- und Einzelbefehlsausführung. Die Adressen für Start, Interrupt-Programme und Stack sind frei wählbar.

Der 6800 hat drei 16 Bit- und drei 8 Bit-Register.
Die Bezeichnungen zeigt Kapitel 7.

- Der Befehlszähler ist 16 Bit lang. Er enthält die jeweils aktuelle Befehlsadresse im laufenden Programm.

- Das Index- und Stack-Register umfaßt ebenfalls 16 Bit. Es läßt sich damit ein Adressenraum von 64 k Bytes überstreichen.

- Die beiden Arbeitsregister (Akkumulatoren) A und B enthalten die Operanden bzw. Ergebnisse der entsprechenden Operationen.

- Das Bedingungsregister kann durch Rechenoperationen oder unmittelbar verändert werden. Es bedeuten:

  N = Negativ
  Z = Null (Zero)
  V = Überlauf (Overflow)
  C = Übertrag (Carry)
  H = Halb-Übertrag (für Dezimalrechnungen)
  I = Interrupt-Mask-Bit.

  Über die bedingten Sprungbefehle lassen sich diese Bits abfragen.

- Die letzten 8 Byte des Speichers enthalten die Startadresse, die Programmanfangsadresse des nicht maskierten Interrupts, des Software-Interrupts und des maskierbaren Hardware-Interrupts.

9. Kurzbeschreibung der Befehle

Der 6800 hat 72 ein-, zwei- oder drei-Byte-Instruktionen. Damit sind Transport-, Setz- und Lösch-Befehle, binäre und dezimale Arithmetik, Shift-, Sprung-, Interrupt- und Stackoperationen möglich.

Es gibt sieben Adressierungsarten:

o Akkumulator (ACCX)

Die die Arbeitsregister A oder B adressierenden Befehle sind ein Byte lang.

o Immediate

In diesen Befehlen dient der im Adressteil stehende Wert selbst als Operand. Es sind zwei oder drei Byte-Befehle.

o Direkt

Mit diesen zwei Byte-Befehlen lassen sich die ersten 256 Bytes des Speichers adressieren.

o Extended

Der Adressteil dieser Befehle besteht aus zwei Byte. Es läßt sich damit auf den Adressenraum von 64 K Byte zugreifen.

o Index

Diese zwei Byte-Befehle bilden vor der Befehlsausführung die neue Adresse aus der Summe des zwei Byte Indexregisters und dem ein Byte Adressteil des Befehls. Dieser Adressteil wird als positive Zahl von 0 bis 255 aufgefaßt.

o Register (Implied)

Diese ein Byte-Befehle adressieren unmittelbar ein oder zwei Register.

o Relativ

Die wirksam werdende Adresse setzt sich aus der Summe des Adressteiles -ein Byte- und des momentanen Befehlszähler-inhaltes zusammen. Damit läßt sich ein Bereich von -125 und + 129 Byte vom eigenen Befehl ausgehend überstreichen. Der Adressteil wird als ganze Zahl in Zweierkomplementdarstellung aufgefaßt.

9.1. <u>Darstellung des Bedingungscodes</u>

Mit einem Byte läßt sich ein Zahlenbereich von 0 bis 255 oder von -128 bis +127 darstellen. Im letzteren Fall werden die negativen Zahlen im sogenannten Zweierkomplement dargestellt. Bit 7 ist dann das Vorzeichen: 0 = plus, 1 = minus.

Führen arithmetische Operationen zum Über- oder Unterschreiten des Zahlenbereichs, so werden automatisch die betreffenden Bits im Bedingungsregister gesetzt. Zum leichteren Verständnis dient Bild 1.

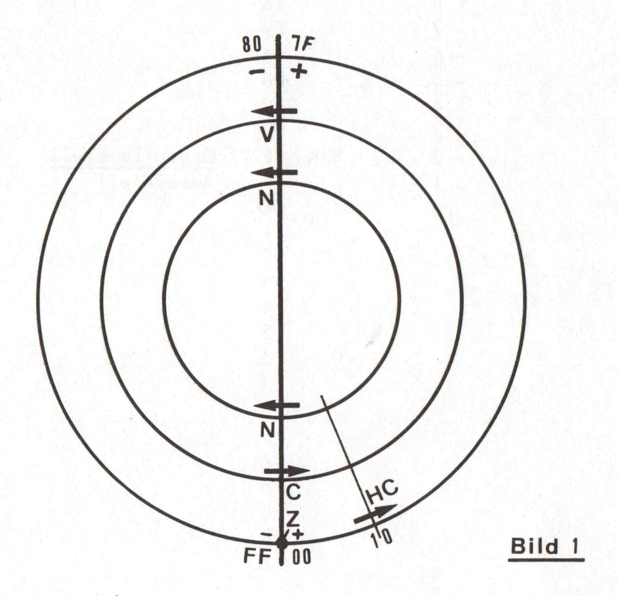


Bild 1

C = Übertrag (Carry)
V = Überlauf (Overflow)
Z = Null (Zero)
N = Negativ
I = Interrupt, Maske für Programmunterbrechung, ist nach Start immer gesetzt
HC = Halb-Übertrag von Bit 3, für dezimales Rechnen

9.2. Tabelle für relative Sprungweiten

Der Bezugspunkt für die Sprungweite ist die OP-Adresse +2

- 11		EF	
- 10		F0	
- F		F1	
- E		F2	
- D		F3	
- C		F4	
- B		F5	
- A		F6	
- 9		F7	
- 8		F8	
- 7		F9	
- 6		FA	
- 5		FB	
- 4		FC	
- 3		FD	
- 2	BRA	FE	Operationsteil
- 1		FF	Adressteil
0		00	
		1	
		2	
		3	
		4	
		5	
		6	
		7	
		8	
		9	
		A	
		B	
		C	
		D	
		E	
		.	
		.	
		.	

Ein Beispiel hierfür befindet sich auf Seite 24.

10. Alphabetische Befehlsliste

ABA	Addiere Akkumulatoren	(Add Accumulators)
ADC	Addiere mit Übertrag	(Add with Carry)
ADD	Addiere	(Add)
AND	Logisches UND	(Logical And)
ASL	Arithmetischer Links-Shift	(Arithmetic Shift Left)
ASR	Arithmetischer Rechts-Shift	(Arithmetic Shift Right)
BCC	Sprung wenn kein Übertrag	(Branch if Carry Clear)
BCS	Sprung wenn Übertrag	(Branch if Carry Set)
BEQ	Sprung wenn Null	(Branch if Equal to Zero)
BGE	Sprung wenn größer gleich Null	(Branch if Greater or Equal Zero)
BGT	Sprung wenn größer Null	(Branch if Greater than Zero)
BHI	Sprung wenn größer	(Branch if Higher)
BIT	Bit Test	(Bit Test)
BLE	Sprung wenn kleiner gleich	(Branch if Less or Equal)
BLS	Sprung wenn kleiner gleich	(Branch if Lower or Same)
BLT	Sprung wenn kleiner Null	(Branch if Less than Zero)
BMI	Sprung wenn Minus	(Branch if Minus)
BNE	Sprung wenn nicht Null	(Branch if Not Equal to Zero)
BPL	Sprung wenn Plus	(Branch if Plus)
BRA	Unbedingter Sprung (relativ)	(Branch Always)
BSR	Unterprogrammsprung (relativ)	(Branch to Subroutine)
BVC	Sprung wenn kein Überlauf	(Branch if Overflow Clear)
BVS	Sprung wenn Überlauf	(Branch if Overflow Set)
CBA	Vergleiche Akkumulatoren	(Compare Accumulators)
CLC	Lösche Übertrag	(Clear Carry)
CLI	Lösche Interrupt-Maske	(Clear Interrupt Mask)
CLR	Lösche	(Clear)
CLV	Lösche Überlauf	(Clear Overflow)
CMP	Vergleiche	(Compare)
COM	Logische Negation	(Complement)
CPX	Vergleiche Indexregister	(Compare Index Register)
DAA	Dezimalangleichung	(Decimal Adjust)
DEC	Vermindere (-1)	(Decrement)
DES	Vermindere Stack-Zähler (-1)	(Decrement Stack Pointer)
DEX	Vermindere Index-Register (-1)	(Decrement Index Register)
EOR	Exklusives ODER	(Exclusive Or)
INC	Erhöhe (+1)	(Increment)
INS	Erhöhe Stack-Zähler (+1)	(Increment Stack Pointer)
INX	Erhöhe Index-Register (+1)	(Increment Index Register)

JMP	Sprung	(Jump)
JSR	Unterprogrammsprung	(Jump to Subroutine)
LDA	Lade Akkumulator	(Load Accumulator)
LDS	Lade Stack-Zähler	(Load Stack Pointer)
LDX	Lade Index-Register	(Load Index Register)
LSR	Logischer Rechtsshift	(Logical Shift Right)
NEG	Arithmetische Negation	(Negate)
NOP	Keine Operation	(No Operation)
ORA	Inklusives ODER	(Inclusive OR Accumulator)
PSH	Transport von Akku. nach Stack	(Push Data)
PUL	Transport von Stack nach Akku.	(Pull Data)
ROL	Kreisshift links	(Rotate Left)
ROR	Kreisshift rechts	(Rotate Right)
RTI	Rücksprung von Interrupt	(Return from Interrupt)
RTS	Rücksprung von Unterprogramm	(Return from Subroutine)
SBA	Subtraktion Akkumulatoren	(Subtract Accumulators)
SBC	Subtraktion mit Übertrag	(Subtract with Carry)
SEC	Setze Übertrag	(Set Carry)
SEI	Setze Interrupt-Maske	(Set Interrupt Mask)
SEV	Setze Überlauf	(Set Overflow)
STA	Speichere Akkumulator	(Store Accumulator)
STS	Speichere Stack Register	(Store Stack Register)
STX	Speichere Indexregister	(Store Index Register)
SUB	Subtrahiere	(Subtract)
SWI	Software-Interrupt	(Software Interrupt)
TAB	Transport Akkumulatoren	(Transfer Accumulators)
TAP	Transport Akku. nach Bed.-Reg.	(Tranf. Accu. to Cond. Reg.)
TBA	Transport Akkumulatoren	(Transfer Accumulators)
TPA	Transport Bed.-Reg. nach Akku.	(Transfer Cond. C. R. to Accu.)
TST	Test	(Test)
TSX	Transport Stack-Register n. Index	(Transfer Stack to Index)
TXS	Transport Index n. Stack-Register	(Transfer Index to Stack)
WAI	Warten auf Interrupt	(Wait for Interrupt)

11. Bedeutung der verwendeten Zeichen und Abkürzungen

OP	Operationscode
B	Byte-Anzahl pro Befehl
Akku A	Akkumulator A
Akuu B	" B
SP	Stackregister
SPH	höherwertiges Byte Stackregister
SPL	niederwertiges Byte Stackregister
X	Indexregister
XH	höherwertiges Byte Indexregister
XL	niederwertiges Byte Indexregister
M	Speicherwort
MSP	Oberstes Speicherwort im Stack
CCR	Bedingungsregister (Condition-Codes-Register)
H	Halbübertrag von Bit 3 (für dezimales Rechnen)
I	Interruptmaske
N	Negativ
Z	Null
V	Überlauf
C	Übertrag von Bit 7
R	wird gelöscht (Reset)
S	wird gesetzt (Set)
/	wird gesetzt wenn Bedingung erfüllt, andernfalls gelöscht
o	bleibt unverändert
00	Byte = Null
→	von - nach
.	Logisches UND
+	Logisches ODER
⊕	Exklusives ODER
-	Logische Negation

Sonderfälle im Bedingungsregister

1	(V)	Test:	Ergebnis = 10000000?
2	(C)	"	Ergebnis = 00000000?
3	(C)	"	höherwertiges BCD-Zeichen größer neun? (wird nicht gelöscht, wenn es vorher gesetzt war)
4	(V)	"	Operand = 10000000 vor der Ausführung?
5	(V)	"	Operand = 01111111 vor der Ausführung?
6	(V)	"	wird auf das Ergebnis von $N \oplus C$ nach Ausführung des Shifts gesetzt
7	(N)	"	Vorzeichenbit des höherwertigen Bytes des Ergebnisses = 1?
8	(V)	"	Überlauf des Zweierkomplements bei der Subtraktion des niederwertigen Bytes?
9	(N)	"	Ergebnis kleiner Null? (Bit 15 = 1)
10	(alle)	"	lädt Bedingungsregister vom Stack
11	(I)	"	wird durch Interrupt gesetzt. Falls vorher gesetzt, kann nur ein nichtmaskierter Interrupt den Wartezustand beenden.
12	(alle)	"	alle Bits werden entsprechend dem Inhalt von Akkumulator A gesetzt.

12. Befehlsliste

12.1. Transportbefehle

Ext. code	Bezeichnung	Wirkung	Immed OP	B	Direkt OP	B	Index OP	B	Extend OP	B	Inher. OP	B	Bedingungsreg. 5 4 3 2 1 0 H I N Z V C
LDAA	Lade Akku A	M→A	86	2	96	2	A6	2	B6	3			o o / / R o
LDAB	Lade Akku B	M→B	C6	2	D6	2	E6	2	F6	3			o o / / R o
LDS	Lade Stack	M→SPH, M+1→SPL	8E	3	9E	2	AE	2	BE	3			o o 9 / R o
LDX	Lade Index	M→XH, M+1→XL	CE	3	DE	2	EE	2	FE	3			o o 9 / R o
PSHA	von Akku A nach Stack	A→MSP, SP-1→SP									36	1	o o o o o o
PSHB	" " B " "	B→MSP, SP-1→SP									37	1	o o o o o o
PULA	vom Stack nach Akku A	SP+1→SP, MSP→A									32	1	o o o o o o
PULB	" " " " B	SP+1→SP, MSP→B									33	1	o o o o o o
STAA	Speichere aus Akku A	A→M			97	2	A7	2	B7	3			o o / / R o
STAB	" " " B	B→M			D7	2	E7	2	F7	3			o o / / R o
STS	" " Stack	SPH→M, SPL→(M+1)			9F	2	AF	2	BF	3			o o 9 / R o
STX	" " Index	XH→M, XL→(M+1)			DF	2	EF	2	FF	3			o o 9 / R o
TAB	Trans. Akku A nach B	A→B									16	1	o o / / R o
TAP	" Akku nach Bed.R.	A→CCR									06	1	----12 ----
TBA	" Akku B nach A	B→A									17	1	o o / / R o
TPA	" Bed. R. nach AkkuA	CCR→A									07	1	o o o o o o
TSX	" Stack nach Index	SP+1→X									30	1	o o o o o o
TXS	" Index nach Stack	X-1→SP									35	1	o o o o o o

12.2. Setz- Lösch- und Shift-Befehle

Ext. code	Bezeichnung	Wirkung	Immed. OP	Immed. B	Direkt OP	Direkt B	Index OP	Index B	Extend OP	Extend B	Inher. OP	Inher. B	Bedingungsreg. 5 4 3 2 1 0 H I N Z V C
CLC	Lösche Übertrag	0→C									0C	1	o o o o o R
CLI	" Interrupt-Maske	0→I									0E	1	o R o o o o
CLV	" Überlauf	0→V									0A	1	o o o o R o
CLR	" Speicher	00→M					6F	2	7F	3			o o R S R R
CLRA	" Akku A	00→A									4F	1	o o R S R R
CLRB	" Akuu B	00→B									5F	1	o o R S R R
SEC	Setze Übertrag	1→C									0D	1	o o o o o S
SEI	" Interrupt-Maske	1→I									0F	1	o S o o o o
SEV	" Überlauf	1→V									0B	1	o o o o S o
ASL	Arith.Shift links Speicher	M: C ← b7…b0 ← 0					68	2	78	3			o o / / 6 /
ASLA	" " " Akku A	A									48	1	o o / / 6 /
ASLB	" " " " B	B									58	1	o o / / 6 /
ASR	" " rechts Speich.	M: b7 → … b0 → C (b7 erhalten)					67	2	77	3			o o / / 6 /
ASRA	" " " Akku A	A									47	1	o o / / 6 /
ASRB	" " " " B	B									57	1	o o / / 6 /
LSR	Logischer Shift rechts Speich.	M: 0 → b7…b0 → C					64	2	74	3			o o R / 6 /
LSRA	" " " Akku A	A									44	1	o o R / 6 /
LSRB	" " " " B	B									54	1	o o R / 6 /
ROL	Kreisshift links, Speicher	M: C ← b7…b0 ← C					69	2	79	3			o o / / 6 /
ROLA	" " Akku A	A									49	1	o o / / 6 /
ROLB	" " " B	B									59	1	o o / / 6 /
ROR	" rechts Speicher	M: C → b7…b0 → C					66	2	76	3			o o / / 6 /
RORA	" " Akku A	A									46	1	o o / / 6 /
RORB	" " " B	B									56	1	o o / / 6 /

12.3 Arithmetische und Logische Befehle

Ext. code	Bezeichnung	Wirkung	Immed. OP	Immed. B	Direkt OP	Direkt B	Index OP	Index B	Extend. OP	Extend. B	Inher. OP	Inher. B	Bedingungsreg. 5 4 3 2 1 0 H I N Z V C
ABA	Addiere Akku A + B	A + B→A									1B	1	/ o / / / /
ADDA	" Sp.+Akku A	A + M→A	8B	2	9B	2	AB	2	BB	3			/ o / / / /
ADDB	" Sp.+Akku B	B + M→B	CB	2	DB	2	EB	2	FB	3			/ o / / / /
ADCA	" Sp.+Ü.+Akku A	A + M + C→A	89	2	99	2	A9	2	B9	3			/ o / / / /
ADCB	" Sp.+Ü+Akku B	B + M + C→B	C9	2	D9	2	E9	2	F9	3			/ o / / / /
SBA	Subtrahiere Akkus	A-B → A									10	1	o o / / / /
SUBA	" aus Akku A	A-M → A	80	2	90	2	A0	2	B0	3			o o / / / /
SUBB	" " " B	B-M → B	C0	2	D0	2	E0	2	F0	3			o o / / / /
SBCA	" mit Übertrag	A-M-C→A	82	2	92	2	A2	2	B2	3			o o / / / /
SBCB	" " "	B-M-C→B	C2	2	D2	2	E2	2	F2	3			o o / / / /
DAA	Dezimalangleichung										19	1	o o / / / 3
DEC	Vermindere Sp.-1	M-1→M					6A	2	7A	3			o o / / 4 o
DECA	" Akku A-1	A-1→A									4A	1	o o / / 4 o
DECB	" Akuu B-1	B-1→B									5A	1	o o / / 4 o
DEX	" Index-1	X-1→X									09	1	o o o / o o
DES	" Stack-1	Sp-1→SP									34	1	o o o o o o
INC	Erhöhe Sp. + 1	M+1→M					6C	2	7C	3			o o / / 5 o
INCA	" Akku A + 1	A+1→A									4C	1	o o / / 5 o
INCB	" Akku B + 1	B+1→B									5C	1	o o / / 5 o
INX	" Index + 1	X+1→X									08	1	o o o / o o
INS	" Stack + 1	SP+1→SP									31	1	o o o o o o

Ext. code	Bezeichnung	Wirkung	Immed. OP	Immed. B	Direkt OP	Direkt B	Index OP	Index B	Extend OP	Extend B	Inher. OP	Inher. B	Bedingungsreg. 5 4 3 2 1 0 H I N Z V C
ANDA	Logisches UND Sp+Akku A	A.M→A	84	2	94	2	A4	2	B4	3			o o / / R o
ANDB	" " " +Akku B	B.M→B	C4	2	D4	2	E4	2	F4	3			o o / / R o
BITA	Bit Test Sp+Akku A	A.M	85	2	95	2	A5	2	B5	3			o o / / R o
BITB	" " " Akku B	B.M	C5	2	D5	2	E5	2	F5	3			o o / / R o
COM	Logische Negation Sp.	$\bar{M}$→M					63	2	73	3			o o / / R S
COMA	" " Akku A	$\bar{A}$→A									43	1	o o / / R S
COMB	" " Akku B	$\bar{B}$→B									53	1	o o / / R S
EORA	Exklusives ODER Akku A	A⊕M→A	88	2	98	2	A8	2	B8	3			o o / / R o
EORB	" " Akku B	B⊕M→B	C8	2	D8	2	E8	2	F8	3			o o / / R o
NEG	Arithmetm. Negation Sp.	00-M→M					60	2	70	3			o o / / 1 2
NEGA	" " Akku A	00-A→A									40	1	o o / / 1 2
NEGB	" " Akku 0	00-B→B									50	1	o o / / 1 2
ORAA	ODER Sp. + Akku A	A+M→A	8A	2	9A	2	AA	2	BA	3			o o / / R o
ORAB	ODER Sp. + Akku B	B+M→B	CA	2	DA	2	EA	2	FA	3			o o / / R o

12.4 Vergleichs- und Sprungbefehle

Ext. code	Bezeichnung	Wirkung	Immed. OP	B	Direkt OP	B	Index OP	B	Extend. OP	B	Inher. OP	B	Bedingungsreg. 5 4 3 2 1 0 H I N Z V C
CBA	Vergleiche Akku	A-B									11	1	o o / / / /
CMPA	" Akku A Sp.	A-M	81	2	91	2	A1	2	B1	3			o o / / / /
CMPB	" Akku B "	B-M	C1	2	D1	2	E1	2	F1	3			o o / / / /
CPX	" Index	XH-M, XL-(M+1)	8C	3	9C	2	AC	2	BC	3			o o 7 / 8 o
TST	Abfrage, Null oder Minus	M-00					6D	2	7D	3			o o / / R R
TSTA	" " " "	A-00									4D	1	o o / / R R
TSTB	" " " "	B-00									5D	1	o o / / R R
BRA	Springe (relativ)	None	20	2									o o o o o o
BCC	" wenn Übertrag = 0	C=0	24	2									o o o o o o
BCS	" " " = 1	C=1	25	2									o o o o o o
BEQ	" " = Null	Z=1	27	2									o o o o o o
BGE	" " ≧ Null	N⊕ V=0	2C	2									o o o o o o
BGT	" " > Null	Z+(N⊕ V)=0	2E	2									o o o o o o
BHI	" " größer	C+Z = 0	22	2									o o o o o o
BLE	" " ≦ Null	Z+(N⊕ V)=1	2F	2									o o o o o o
BLS	" " kleiner oder gleich	C+Z=1	23	2									o o o o o o
BLT	" " < Null	N⊕V=1	2D	2									o o o o o o
BMI	" " Minus	N=1	2B	2									o o o o o o
BNE	" " nicht Null	Z=0	26	2									o o o o o o
BVC	" " Überlauf=0	V=0	28	2									o o o o o o
BVS	" " " =1	V=1	29	2									o o o o o o
BPL	" " Plus	N=0	2A	2									o o o o o o
BSR	" Unterprogramm relat.		8D	2									o o o o o o
JMP	Springe Absolut	Siehe Spezial-					6E	2	7E	3			o o o o o o
JSR	" Unterprogramm Abs.	operationen.					AD	2	BD	3			o o o o o o
NOP	Keine Operation										02	1	o o o o o o
RTI	Rücksprung von Interrupt	Siehe Spezial-									3B	1	----10-----
RTS	" " Unterprogramm	operationen.									39	1	o o o o o o
SWI	Software-Interrupt										3F	1	o S o o o o
WAI	Warten auf Interrupt										3E	1	o 11o o o o

13. Wirkung von Spezialoperationen

JSR - Unterprogrammsprung:

INDXD

PC	Hauptprogramm
n	AD = JSR
n + 1	K = Offset *
n + 2	Nächster Befehl

* K = 8-Bit Wert ohne Vorzeichen

SP	Stack
→ SP-2	
SP-1	(n + 2) H
SP	(n + 2) L

$(n + 2)_H$ und $(n + 2)_L$ bilden n + 2

PC	Unterprogramm
X + K	1. Unterpr.-Befehl

EXTND

PC	Hauptprogramm
n	BD = JSR
n + 1	SH = Unterpr.Adr.
n + 2	SL = Unterpr.Adr.
n + 3	Nächster Befehl

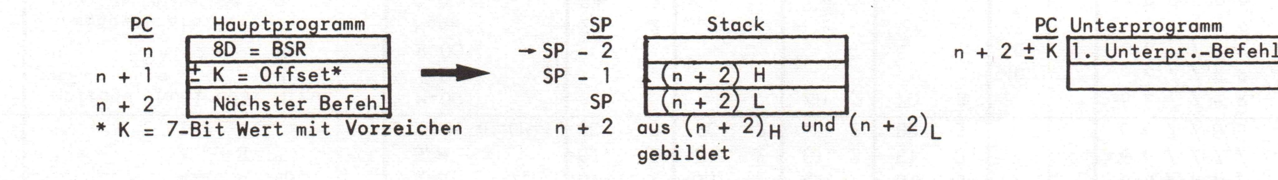
SP	Stack
→ SP- 2	
SP- 1	(n + 3) H
SP	(n + 3) L

→ = Stackregister nach Ausführung

PC	Unterprogramm
S	1. Unterpr.-Befehl

(S aus S_H und S_L gebildet)

BSR - Unterprogrammsprung (relativ:)



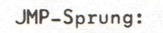
PC	Hauptprogramm
n	8D = BSR
n + 1	± K = Offset*
n + 2	Nächster Befehl

* K = 7-Bit Wert mit Vorzeichen

SP	Stack
→ SP - 2	
SP - 1	(n + 2) H
SP	(n + 2) L

n + 2 aus $(n + 2)_H$ und $(n + 2)_L$ gebildet

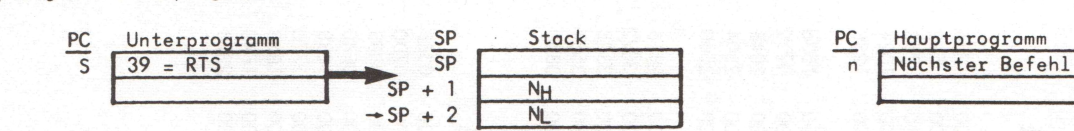
PC	Unterprogramm
n + 2 ± K	1. Unterpr.-Befehl



JMP-Sprung:



RTS - Rücksprung von Unterprogramm:



14. Konvertierungstafeln, Potenzen von 2

Zahlenumrechnung

hexadezimal - dezimal

1000	4096	0100	256	0010	16	0001	1
2000	8192	0200	512	0020	32	0002	2
3000	12288	0300	768	0030	48	0003	3
4000	16384	0400	1024	0040	64	0004	4
5000	20480	0500	1280	0050	80	0005	5
6000	24576	0600	1536	0060	96	0006	6
7000	28672	0700	1792	0070	112	0007	7
8000	32768	0800	2048	0080	128	0008	8
9000	36864	0900	2304	0090	144	0009	9
A000	40960	0A00	2560	00A0	160	000A	10
B000	45056	0B00	2816	00B0	176	000B	11
C000	49152	0C00	3072	00C0	192	000C	12
D000	53248	0D00	3328	00D0	208	000D	13
E000	57344	0E00	3584	00E0	224	000E	14
F000	61440	0F00	3840	00F0	240	000F	15

dezimal - hexadezimal

10000	2710	1000	03E8	100	0064	10	000A
20000	4E20	2000	07D0	200	00C8	20	0014
30000	7530	3000	0BB8	300	012C	30	001E
40000	9C40	4000	0FA0	400	0190	40	0028
50000	C350	5000	1388	500	01F4	50	0032
60000	EA60	6000	1770	600	0258	60	003C
		7000	1B58	700	02BC	70	0046
		8000	1F40	800	0320	80	0050
		9000	2328	900	0384	90	005A

dezimal-hexadezimal-dual

0	0	0 0000
1	1	0 0001
2	2	0 0010
3	3	0 0011
4	4	0 0100
5	5	0 0101
6	6	0 0110
7	7	0 0111
8	8	0 1000
9	9	0 1001
10	A	0 1010
11	B	0 1011
12	C	0 1100
13	D	0 1101
14	E	0 1110
15	F	0 1111
16	10	1 0000
17	11	1 0001
18	12	1 0010
19	13	1 0011
20	14	1 0100

Potenzen von 2

0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536
17	131072
18	262144
19	524288
20	1048576

15. Stichworte: englisch - deutsch

Begriff	Übersetzung	Erläuterung
Access	Zugriff	Möglichkeit, eine bestimmte Speicherzelle anzusprechen.
Accumulator	Akkumulator	Rechenfähiges Register
Add	addiere	
Address	Adresse	Speicherstelle
Adjust	anpassen	
Always	immer	
And	und	
Bit		(Binary Digit)binäre Informationseinheit
Branch	Sprung, Verzweigung	
Byte		Kleinste adressierbare Einheit 8 Bit = 1 Byte.
Carry	Übertrag	
Clear	löschen	
Compare	vergleichen	
Complement	Ergänzung	
Condition	Bedingung	
Data	Daten	
Decrement	vermindern, abziehen	
Equal	gleich	
Execute	ausführen	
Greater	größer	
Half-Carry	Halb-Übertrag	für Dezimalrechnungen
Higher	höher, größer	

If	wenn, falls	
Immediate	unmittelbar	Befehlsart in der der im Adressteil stehende Wert als Operand dient.
Interrupt	Unterbrechung	Unterbrechung eines laufenden Programmes und Start einer Interruptroutine. Anschließend Fortsetzung des unterbrochenen Programmes.
Instruction	Befehl	Anweisung für den Computer.
Jump	Sprung	
Left	links	
Less	weniger	
Load	laden	
Lower	tiefer, weniger	
Main Routine	Hauptprogramm	
Memory	Speicher	
Negate	negieren	
Or	oder	
Overflow	Überlauf	
Pointer	Zähler, Zeiger	
Pull	ziehen	Transport von Akkumulator nach Stack.
Push	drücken	Transport nach Akkumulator von Stack.
RAM		Random-Access-Memory. Speicher mit wahlfreiem Zugriff.
Reset	zurücksetzen	
Result	Ergebnis, Resultat	

Return	zurückkehren	
Right	rechts	
Rotate	rotieren	bei Shift = Kreisshift.
Same	dasselbe	
Set	setzen	
Shift	schieben	
Signed	mit Vorzeichen	
Special Operations	Spezialoperationen	
Stack	Kellerspeicher	
Statement	Befehl	siehe auch Instruction.
Storage	Speicher	
Store	speichern	
Subroutine	Unterprogramm	
Subtract	Subtrahieren	
Transfer	transportiere	
Unsigned	ohne Vorzeichen	Absolutwert
Value	Wert	
Wait	warten	
Zero	Null	

16. Kassettenrekorderanschluß

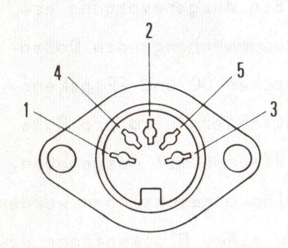
16.1. Einführung

Mit dem Kassettenanschluß bietet das TV-Computersystem die Möglichkeit der Ein- und Ausgabe über handelsübliche Kassettenrecorder. Die externe Speicherung von Programmen auf Kassetten vereinfacht das Arbeiten mit dem System erheblich. Die Kassetten-Ein- Ausgabe von beliebigen Speicherbereichen erspart die wiederholte Eingabe von Programmen bzw. Daten per Lichtgriffel.

In der Programmentwicklungsphase wird man den jeweils aktuellen Stand oder auch mehrere Versionen eines Programms auf einem Arbeitsband aufzeichnen. Ausgetestete Programme können auf einem Bibliotheksband zur Verfügung stehen.

16.1.1. Anschlußbedingungen

Das Aufzeichnungsverfahren stellt keine besonderen Anforderungen an die Kassettenrecorder. Als Anhaltspunkt sei erwähnt, daß das System mit einem preiswerten Gerät, das von der Stiftung Warentest in der Zeitschrift Test 7/76 mit "zufriedenstellend" bewertet wurde, einwandfrei funktioniert.



pin 2 : Masse
pin 1 : Ausgang
pin 3 : Eingang

Bild 16.1.1.

16.1.2. Arbeitsweise der Kassetten- E/A

Die Ein- und Ausgabe wird per Mikroprozessor durch zwei in einem Festwertspeicher - 1/2 K PROM im Adressbereich C000 bis C1FF - gespeicherte Programme gesteuert. Alle variablen Daten dieser Programme liegen auf Seite 1F im Adressbereich von 3E0 bis 3F5. Es sind dies vom Benutzer bereitzustellende Parameter, Hilfszellen der Programme und der Stack. Der übrige Speicher ist für den Benutzer frei verfügbar.

Das Programm darf während des Ausgabe- bzw. Eingabevorgangs nicht unterbrochen werden, da das Erzeugen bzw. Erkennen der verschiedenen Impulsfrequenzen zeitkritische Vorgänge sind. Im Normalbetrieb rechnet der Prozessor pro Bildwiederholung nur in der Zeit zwischen unterem Bildende und Bildwechsel während der Bildschirmanzeige des Speichers wird er gestoppt. Aus den genannten Gründen muß also die Speicherabbildung auf dem Fernsehschirm während der Datenübertragung zum bzw. vom Kassettengerät unterbrochen werden. Der Bildschirm bleibt bis auf die Steuerfelder am rechten Rand in dieser Zeit dunkel.

Durch einen Start des Kassetten-Ausgabeprogramms wird ein beliebig langer,zusammenhängender Speicherbereich, ergänzt um einige organisatorische Daten, auf Band geschrieben. Ein Ausgabevorgang erzeugt auf dem Band einen variabel langen,zusammenhängenden Datenblock, der durch eine Identitätsnummer zwischen 00 und FF gekennzeichnet ist. Zum Erkennen von Übertragungsfehlern wird pro Byte ein Paritätsbit und pro Block eine Längsprüfsumme mit ausgegeben. Auf ein Band können nacheinander mehrere Blöcke geschrieben werden. Eine spezielle Synchronisation zum Erkennen eines Blockanfangs erlaubt es, zwischen Datenblöcken Sprache (oder Musik) aufzuzeichnen. Wir empfehlen, vor jedem Datenblock seine wichtigsten Angaben auf Band zu sprechen, so daß das Band gleichzeitig ein "verständliches" Inhaltsverzeichnis enthält.

Bei der Eingabe wird der durch die vorgegebene Identitätsnummer gekennzeichnete Datenblock auf Band gesucht und - falls vorhanden - in den Speicher eingelesen. Dabei muß das Band nicht unmittelbar vor dem gesuchten Datenblock stehen, da Blöcke mit ungleicher Identnummer (und Sprache) beim Suchvorgang überlesen werden. Das Eingabeprogramm nimmt eine Paritykontrolle vor und meldet gegebenenfalls einen Übertragungsfehler.

16.1.3. Angaben zur Zeitabschätzung

Vor jedem Datenblock wird ein Vorspann ausgegeben, der zur Synchronisierung und zur Identifizierung des Blocks dient. Für die Ausgabe des Vorspanns sind ca. 7,8 sec. zu veranschlagen. Die Ausgabe eines Bytes dauert ca. 75 msec. Damit ergeben sich folgende Übertragungszeiten:

1 TV-Seite	(32 Bytes)	ca. 10 sec
1/2 K	(512 Bytes)	ca. 46 sec
1 K	(1024 Bytes)	ca. 85 sec.

16.1.4. Erläuterung der Parameter

Identitätsnummer

Die Identitätsnummer ist im Bereich von 00 bis FF wählbar. Der Benutzer sollte darauf achten, daß eine Nummer pro Band nur einmal vergeben wird.

Anfangsadresse

Die Anfangsadresse (2 Bytes) ist bei der Ausgabe stets erforderlich, bei der Eingabe nur bei verschieblichem Format.

Endadresse + 1

Die um eins erhöhte Endadresse ist nur als Ausgabeparameter erforderlich.

Modus

Dieser Parameter unterscheidet die zwei Fälle:

Modus = 0: absolutes Programm

Modus ≠ 0: verschiebliches Programm.

Absolut bedeutet, daß ein Programm (z. B. durch Adressen) an einen festen Speicherbereich gebunden ist. Bei der Ausgabe eines absolut gekennzeichneten Speicherbereichs wird die Anfangsadresse mit auf Band geschrieben. Bei der Eingabe wird der Block in den ursprünglichen Bereich geladen, die Angabe einer Anfangsadresse als Parameter erübrigt sich.

Der Inhalt eines verschieblich gekennzeichneten Bereichs kann an jede beliebige Stelle eingelesen werden. Hierzu ist bei der Eingabe die Anfangsadresse als Parameter bereitzustellen.

Die Modusangabe wird vom Ein- und Ausgabeprogramm abgefragt. Dabei können widersprüchliche Angaben auftreten. Die vier möglichen Kombinationen sind:

Ausgabemodus	Eingabemodus	Bemerkung
absolut	absolut	-
absolut	verschieblich	siehe 1)
verschieblich	verschieblich	-
verschieblich	absolut	siehe 2)

1) In diesem Fall geben wir bei der Eingabe dem Modus verschieblich den Vorrang. Das heißt, die auf Band gespeicherte Anfangsadresse wird ignoriert, der Block wird ab der in den Zellen 3E2, 3E3 gespeicherten Anfangsadresse abgelegt.

2) Dieser Fall führt zu einer Fehlermeldung (Rückmeldung FE) und zum Abbruch des Eingabeprogramms. Da bei verschieblichem Ausgabemodus keine Adresse aufgezeichnet wird und beim absoluten Eingabemodus keine Adresse bereitzustellen ist, kann der Block nicht geladen werden.

16.2. Bedienung des Kassetten-Ausgabeprogramms

Die Startadresse des Ausgabeprogramms ist C000. Das Programm erwartet vier Parameter:

- Identitätsnummer
- Anfangsadresse
- Endadresse + 1
- Modusangabe

Adresse	
3E0	Identnr.
3E2	Anfangsadresse
3E4	Endadresse + 1
3E6	Modus
.	
.	
.	
3FE	Startadresse

Bild 16.2.1. Parameter des Kassettenausgabeprogramms auf Seite 1F

Bei der Ausgabe eines Speicherbereichs ist in folgenden Schritten vorzugehen:

- Eintragen der Ausgabeparameter auf Seite 1F (siehe Bild 16.2.1.)

  - Adresse 3E1: Identitätsnummer
  - Adresse 3E2, 3E3: Anfangsadresse (höherwertiges Byte in 3E2, niederwertiges Byte in 3E3)
  - Adresse 3E4, 3E5: Endadresse + 1
  - Adresse 3E6: Modus

- Eintragen der Startadresse C000 in der letzten Zeile von Seite 1F.

- Vorbereiten und Anschließen des Kassettengeräts, Band zurückspulen oder nach letztem Datenblock positionieren, Lautstärkeregler etwa in mittlerer Position einstellen.

- Start des Kassettengeräts im Aufnahmemodus

- Nach einigen Sekunden Start des Mikroprozessors

Unmittelbar nach dem Start des Ausgabeprogramms wird, wie bereits erwähnt, die Speicheranzeige dunkel. Das Wiederaufleuchten des Bildschirms zeigt das Ende des Übertragungsvorgangs an.

- Nach einigen Sekunden Band stoppen.
- Mikroprozessor stoppen.

Für jedes Band sollte ein Inhaltsverzeichnis geführt und bei jeder Ausgabe auf den aktuellen Stand gebracht werden.

Beispiel:

Ein absolutes Programm, das die Seiten 1 bis 5 belegt, soll unter der Identitätsnummer 33 ausgegeben werden. Bild 16.2.2 zeigt die Parameter auf Seite 1F. Man beachte vor allem, daß die um eins erhöhte Endadresse einzutragen ist.

Franz MORAT KG
Elektro-Feinmechanik und Maschinenbau
7821 Eisenbach Hochschwarzwald 1

Fernruf
Eisenbach
(0 76 57) * 444, 292

Fernschreiber
07 722 323
fmkg d

TV-Computersystem 6800

Seite: 1F

Aufgabe: Beispiel: Kassettenausgabe

Nr.	Byte	Marke	OPEX	A	Symb. Adr.	CODE	Kommentar
0	3E0					0 0	Indentitätsnummer
1	1					3 3	
2	2					0 0	Anfangsadresse des auszugebenden
3	3					2 0	Programms
4	4					0 0	Endadresse +1 des auszugebenden
5	5					C 0	Programms
6	6					0 0	Modus: absolutes Programm
7	7						
8	8						
9	9						
10	A						
11	B						
12	C						
13	D						
14	E						
15	F						
16	3F0						
17	11						
18	12						
19	13						
20	14						
21	15						
22	16						
23	17						
24	18						
25	19						
26	1A						
27	1B						
28	1C						
29	1D						
30	1E					C 0	Startadresse des Kassetten-
31	1F					0 0	ausgabeprogramms

A = Adressenmodus
CODE = interner Code

A: Immediate = I, Implied = M,
Extended = E, Relativ = R,
Index = X, Accu. = A oder B

Bild 16.2.2.

16.3. Bedienung des Kassetten-Eingabeprogramms

Die Startadresse des Eingabeprogramms ist C100. Parameter sind:

- Indentitätsnummer
- Modus
- Anfangsadresse, falls Modus = verschieblich.

Das Programm liefert eine Rückmeldung unter Adresse 3E0.

Adresse		
3E0	Rückm.	Identnr.
3E2	(Anfangsadresse)	
3E4		
3E6	Modus	
.		
.		
.		
3EF	Startadresse	

Bild 16.3.1. Parameter des Kassetten-Eingabeprogramms

Bei der Kassetten-Eingabe ist in folgenden Schritten vorzugehen:

- Eintragen der Eingabeparameter auf Seite 1F (siehe Bild 16.3.1)
  - Adresse 3E1: Identitätsnummer
  - Adresse 3E6: Modus (=0 bedeutet absolut, ≠0 verschieblich)
  - falls Modus ≠0:
    Adresse 3E2, 3E3: Anfangsadresse.

- Eintragen der Startadresse C100 unter letzten Zeile von Seite 1F.

- Vorbereiten und Anschließen des Kassettengerätes. Das Band entweder an den Anfang zurückspulen und Block suchen lassen oder

Band vor dem betreffenden Block anhalten. Lautstärkeregler in mittlerer Position einstellen.

- Start des Mikroprozessors

Start des Kassettenrekorders in Wiedergabemodus.

Wie bei der Ausgabe ist der Bildschirm auch während des Eingabevorgangs dunkel. Das Wiederaufleuchten des Bildschirms zeigt das Ende der Übertragung an. Wird die vorgegebene Identitätsnummer nicht gefunden, läuft das Band immer weiter und auch der Bildschirm bleibt dunkel. Um diese Situation zu erkennen, ist das Abschätzen der Eingabezeit nützlich.

- Band und Mikroprozessor stoppen.

Das Eingabeprogramm liefert im Byte 3E0 eine Rückmeldung, die drei verschiedene Werte annehmen kann:

Rückmeldung = 00: Gesuchter Datenblock wurde fehlerfrei in den Speicher eingelesen.

" = FF: Es trat ein Übertragungsfehler (Parityfehler) auf, die Übertragung wurde an der betreffenden Stelle abgebrochen und damit nicht ordnungsgemäß abgeschlossen. Die Eingabe sollte nochmals versucht werden.

" = FE: Benutzerfehler durch widersprüchliche Modusangaben bei Ein- und Ausgabe, siehe 16.1.4.

17. Bus-Anschluß

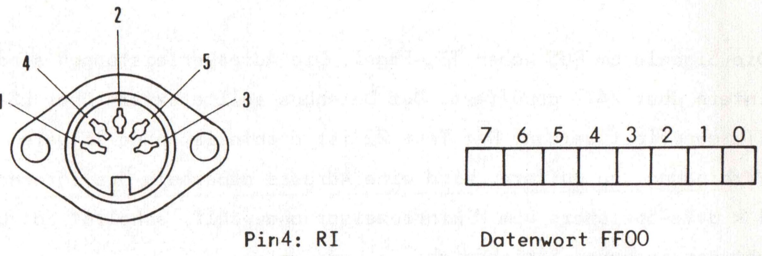
Die Signale am BUS haben TTL-Pegel. Die Adressenleistungen sind intern über 7417 gepuffert. Der Datenbus sollte extern über LS-TTL gepuffert werden. Der Takt Ø2 ist ebenfalls, ohne längere Verbindung, zu puffern. Wird eine Adresse außerhalb des internen 1 k Byte-Speichers vom Mikroprozessor angewählt, schaltet das Gerät den internen Speicher ab.

Das Signal gibt die Zeit an, in der der Mikroprozessor auf den Speicher des TV-Systems zugreifen kann. Dies sind alle 18 µsec. je 2 µsec. Während der 18 µsec. greift die Anzeigeelektronik auf dem Speicher zu. In dieser Zeit befindet sich der 6800 im Haltzustand.

18. Relaisanschluß

Über die Anschlüsse 4 und 5 der 5-poligen Tonabnehmerbuchse (Mab5S) ist jeweils ein Relais per Programm ansteuerbar. Wir empfehlen und liefern R-Relais der Fa. National (165 Ohm) mit einem Umschaltkontakt. Die Wicklung wird unmittelbar zwischen die Anschlußstifte und + 5V geschaltet.

Die Relais sind über das Datenwort FF00 Bit 0 und/oder Bit 1 anzusprechen. Über Bit 2 läßt sich die Speicherausgabe unterdrükken. Bit 7 enthält das Datenbit der Kassetten-Ein- und Ausgabe.

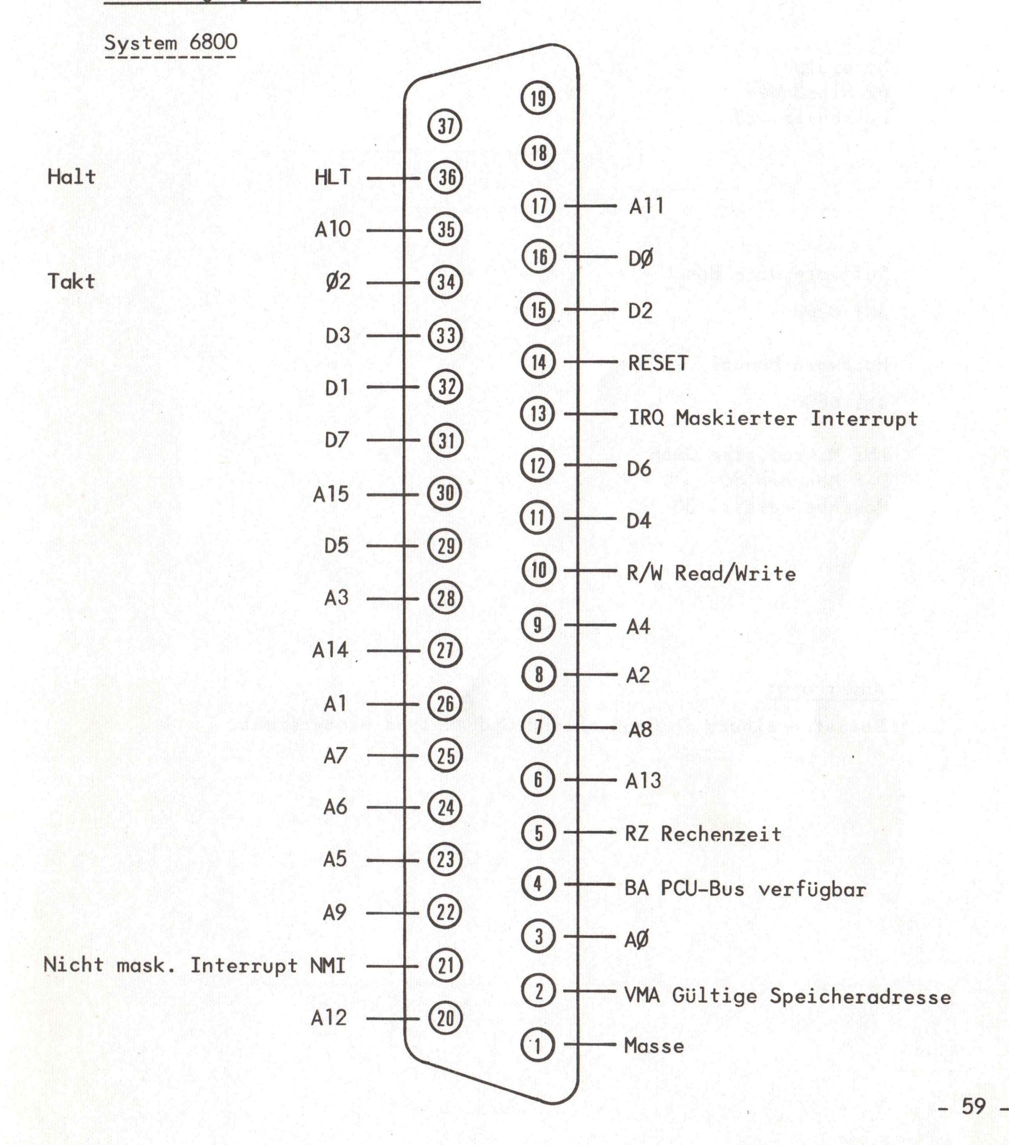


Mab5S

Das Relais Bit 0 wird durch das Kassettenprogramm angesteuert und ist daher bei der Kassetten-Ein- und Ausgabe nicht frei verfügbar.

Pin-Belegung des Busanschlusses

System 6800



19. Literaturhinweise

M 6800 Mikroprozessor
Programming Manual — Motorola

M 6800 Mikroprozessor
Applications Manual — Motorola

Motorola
62 Wiesbaden
Luisenstr. 28

Software Data Book

AMI 6800

Hardware Manual

AMI 6800

AMI Mikrosystem GmbH
D-8 München 80
Rosenheimerstr. 30

Anmerkung:

Patentanmeldung in Deutschland und Ausland eingereicht.

Speicherseite, 32 Bytes oder 256 Bit

Bedienungsfelder

Seite mit angewählter Tetrade setzen

Seitendarstellung Bit/Tetrade

Seitenzähler vorw.

Ausgewählte Tetrade

Seitenzähler rückw.

Setzen, Anzeige } höherwertige Stelle

Setzen, Anzeige } niederwertige Stelle

} Seitenzähler 32 Seiten ≙ 1 k Bytes

Interrupt

Einzelbefehl mit Registeranzeige

Halt/Fortstart

Grundzustand bzw. Start

} Mikroprozessorsteuerung

Steuerfelder für Lichtgriffel

Anzeige Schalterzustand

Tetradenauswahl

Haltenorth